Data Science Capstone project

<Khayom Mirzoaminov>

<August 17, 2021>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
- We have collected data to predict the landing outcomes from SpaceX

- Summary of all results
- As result the R square of our model was 83% which is quite good.

Introduction

- Project background and context
- SpaceX are designing multi-usable rocket to reduce the cost of space travel. We predict if the Falcon 9 first stage will land successfully or not.

- Problems you want to find answers.
- Should we use data from other companies to improve the model?
- What are the reasons of unsuccessful landing.

Methodology

- Data collection methodology:
  - Describe how data were collected
  - We get data from SpaceX API and Wikipedia.
  - Clean it for further analyze using Pandas, Request and BeautifulSoup modules.
- Perform data wrangling
  - Describe how data were processed
  - We have import Pandas and Numpy. We have calculated the number of launches on each site, number and occurrence of each orbit and mission outcome per orbit type.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - We use SQL and Visualization to find pattern of correlation. SQL Help us to store the data set and plot variables to see how they effect on each other. For example, the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return
- Perform interactive visual analytics using Folium and Plotly Dash
  - This packges helps us to generated map with marked launch sites and understand the following questions Are all launch sites in proximity to the Equator line? Also we have mark the success/failed launches for each site on the map. We discovered many interesting insights related to the launch sites' location using folium, in a very interactive way. With Ploty Dash, we enhance our findings.
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Methodology

Data collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data collection – SpaceX API

https://github.com/mkhayom/master/blob/master/W1_Collection%20API.ipynb

Show the summary of the dataframe

[28]:
```
# Show the head of the dataframe
df.head()
```

[28]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Bloc
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	Nal
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	Nal
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	Nal
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	Nal
		2010				CCSFS SLC	None						

Data collection – Web scraping

https://github.com/mkhayom/master/blob/master/W1_jupyter-labs-webscraping.ipynb

```
]: df=pd.DataFrame(launch_dict)
   df
```

]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Data wrangling

- Describe how data were processed
  - Find all missing values and replaced by Mean. Find success rate by deterring outcome
- You need to present your data wrangling process using key phrases and flowcharts
  - `for i,outcome in enumerate(landing_outcomes.keys()): print(i,outcome)`
  - bad_outcomes {'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- https://github.com/mkhayom/master/blob/master/W1_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with data visualization

- Summarize what charts were plotted and why used those charts
  - We have plot Bar, Scatter and Line char to see how different variable correlate with each other. For example, we can see that ES_L1, GEO, HEO and SSO orbits have high success rate.
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/mkhayom/master/blob/master/W2_jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Summarize performed SQL queries using bullet points
  - We find the unique launch sites in the space mission
  - We find total payload mass carried by boosters launched by customer
  - Got average payload mass carried by booster version F9 v1.1
  - Listed the names of the boosters which have success in drone ship by multiple condition clause.
  - Use a subquery
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
  - https://github.com/mkhayom/master/blob/master/W2_EDA%20with%20SQL.ipynb

Build an interactive map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - I have marked all launch sites on a map, also add markers with outcome of each flight. Calculate distance between different objects.
- Explain why you added those objects
  - Putting outcome result on location of each launching pad gives me descriptive and visual view on success rate.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
  - https://github.com/mkhayom/master/blob/master/W3_lab_jupyter_launch_site_location.ipynb

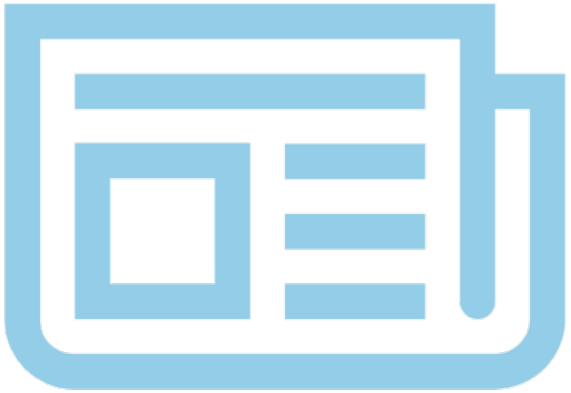
Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - I have created chart and scatter plots. As variable I have used Launchpad site, class outcome, booster version and payload range.
- Explain why you added those plots and interactions
  - With dashboard we identify that followings have high success rate:
    - “FT” Booster version
    - KSC LC-39A Launch site
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
  - https://github.com/mkhayom/master/blob/master/spacex_dash_app.py

Predictive analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- Standardized the data. Split the data to test and training set.
- Then created logistic regression, support vector machine, decision tree classifier and k nearest neighbors.
- Calculated R square for each model and confusion matrix.

Results



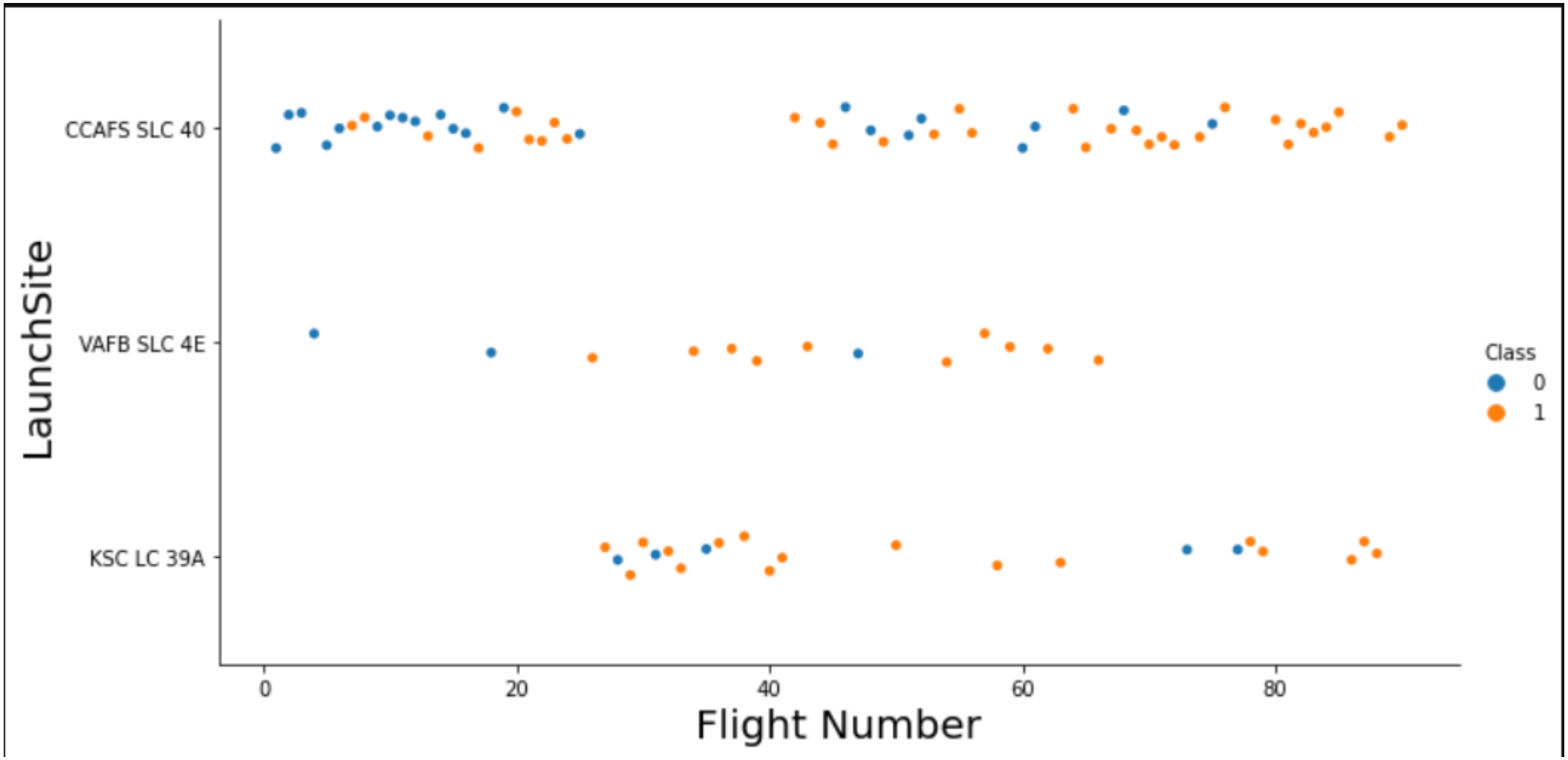
- Exploratory data analysis results
- As result we have two dataframe ready for further analyses.
- Interactive analytics demo in screenshots

- Predictive analysis results

EDA with Visualization

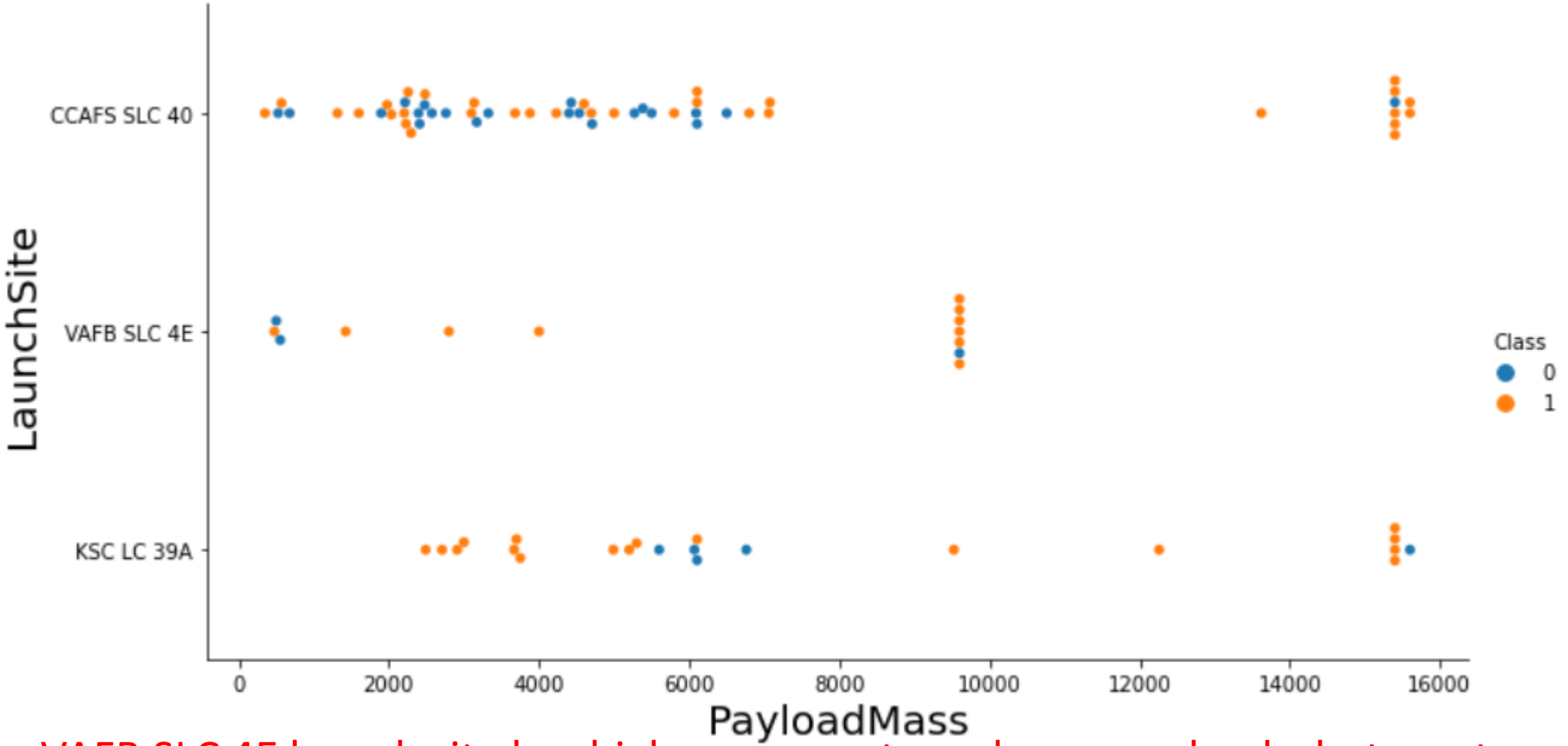
https://github.com/mkhayom/master/blob/master/W2_jupyter-labs-eda-dataviz.ipynb

Flight Number vs. Launch Site



Most flight were on CCAFS SLC 40 launch site and KSC LC 39A launch site has high success rate.

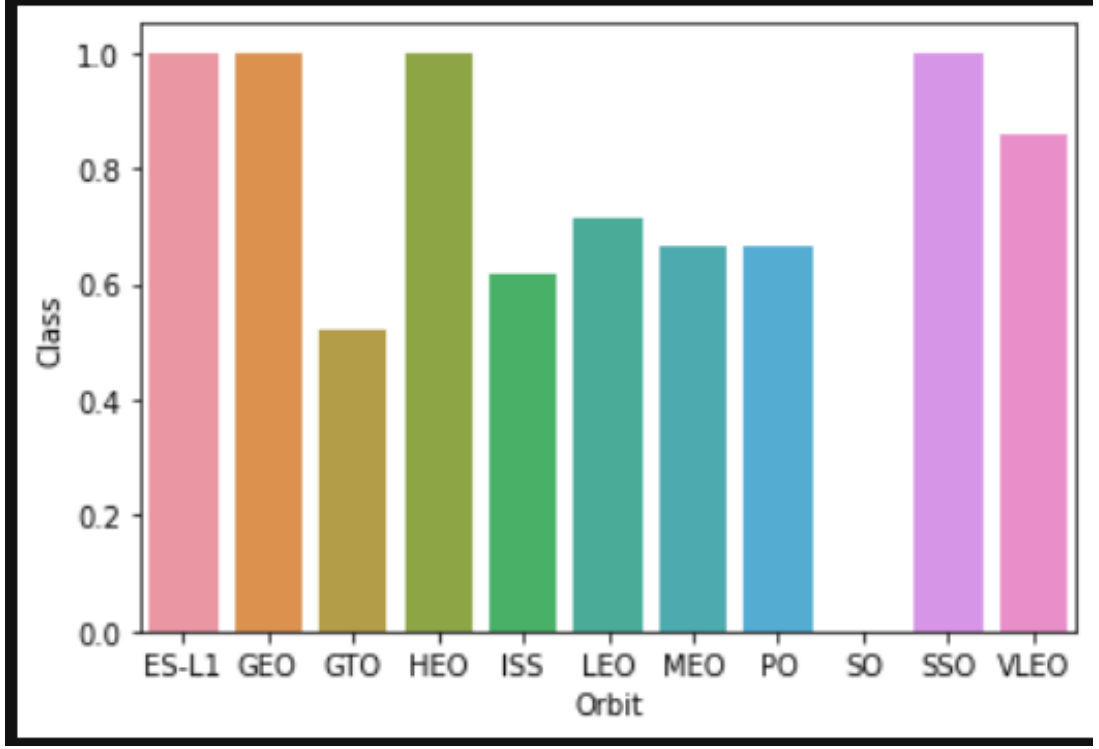
Payload vs. Launch Site



VAFB SLC 4E launch site has high success rate on heavy payloads, but most flight conducted with payload in range of 2000 – 7000

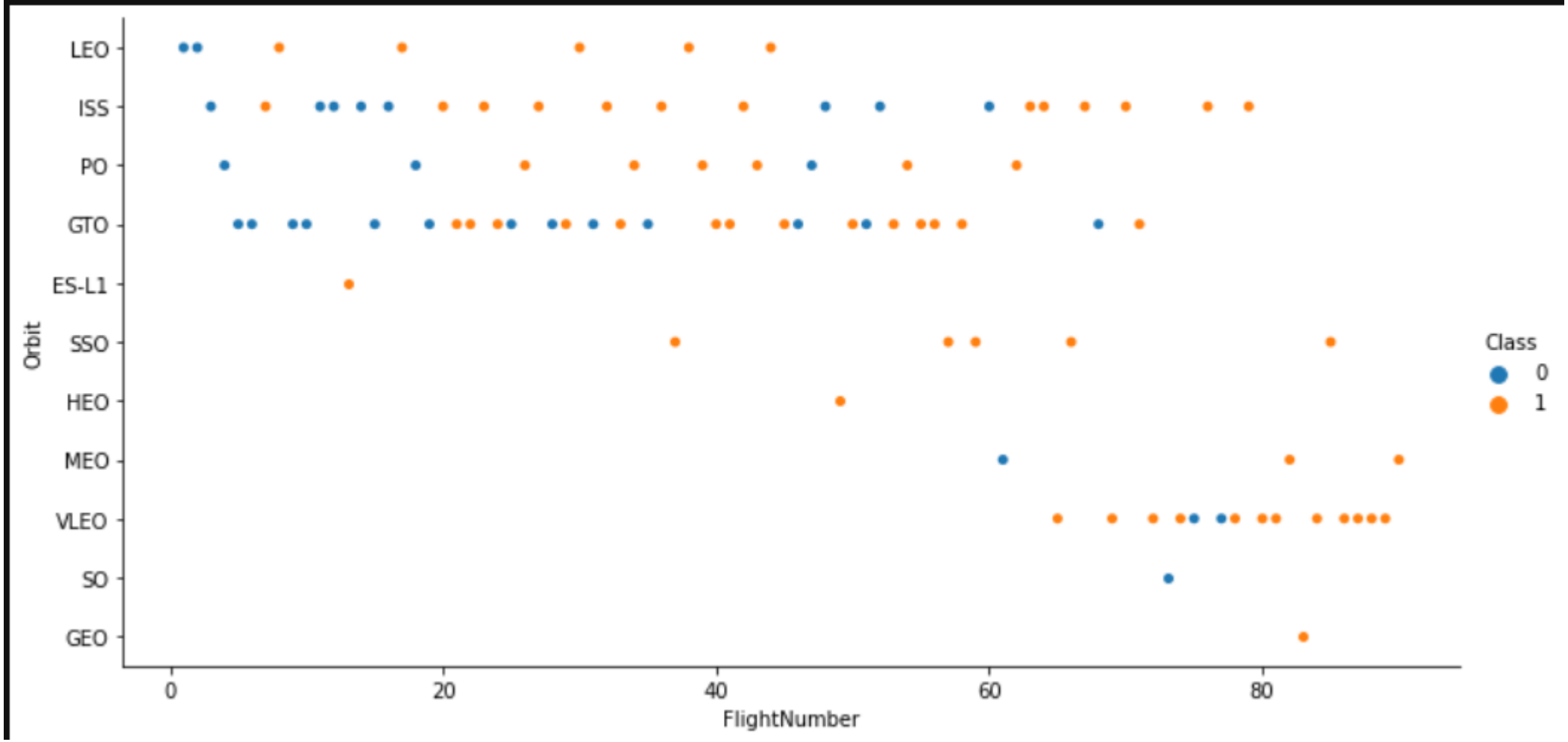
Success rate vs. Orbit type

```
meandf = df.groupby('Orbit').Class.mean().reset_index
sns.barplot(y='Class', x='Orbit',data=meandf)
plt.show()
```



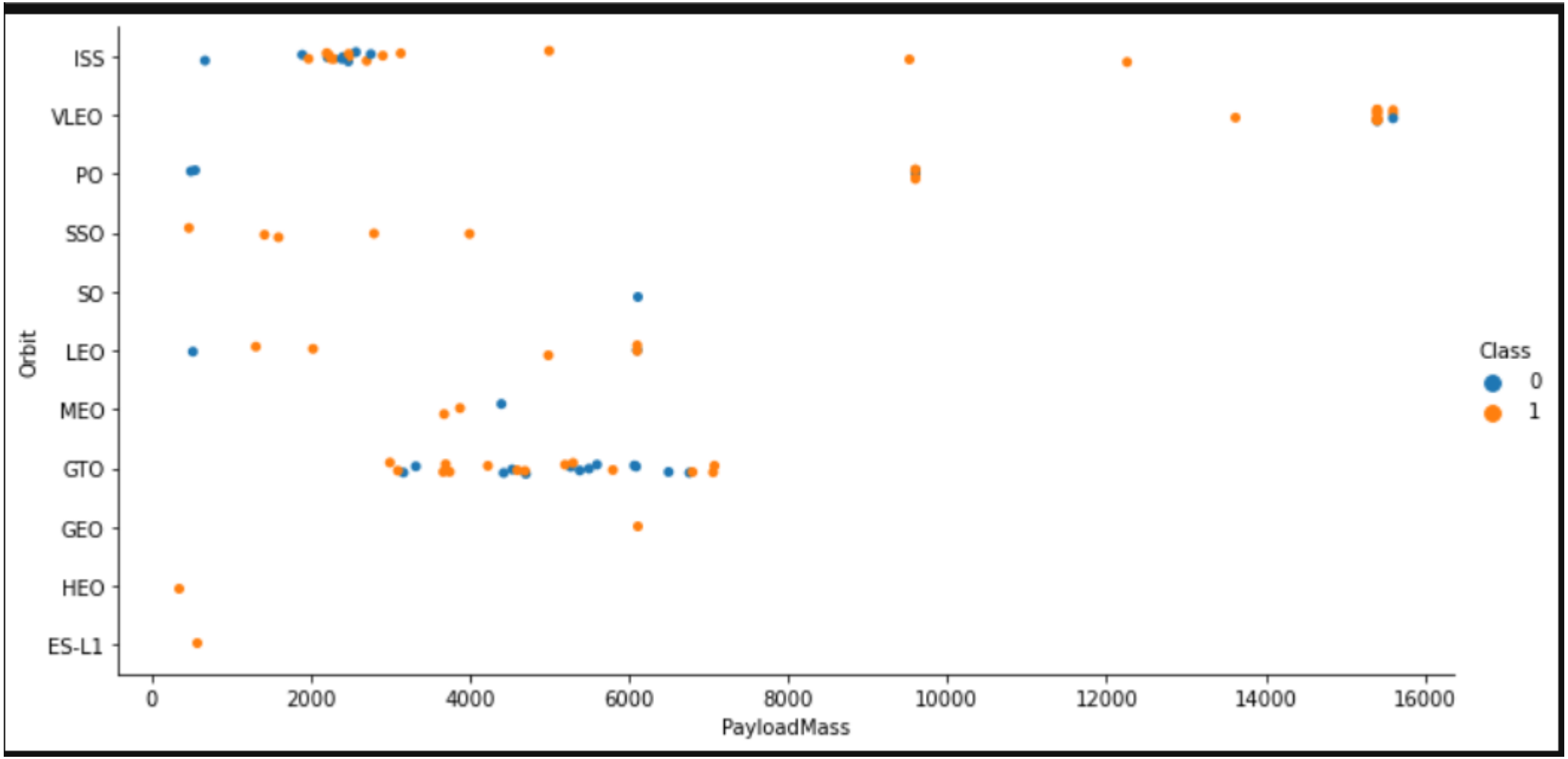
- We see that higher earth orbit have high success rate – ESL1, GEO, GTO, HEO

Flight Number vs. Orbit type



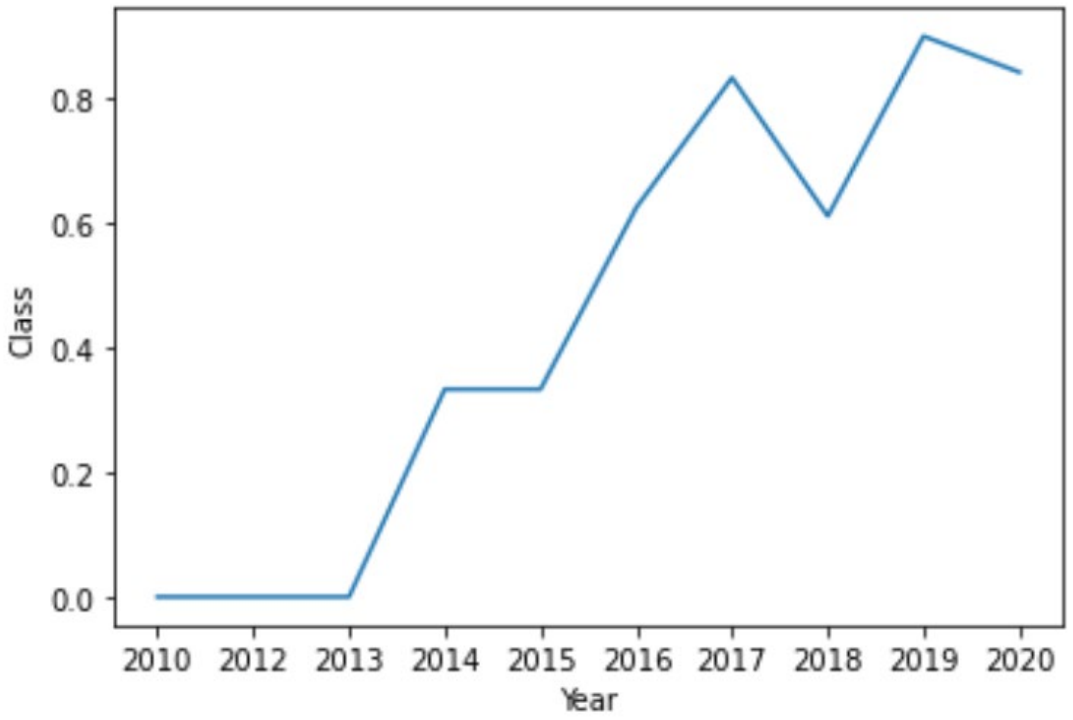
We see that higher earth orbit have high success rate – ESL1, GEO, GTO, HEO. But most flight were conducted in low orbits like – ISS, VLEO, LEO

Payload vs. Orbit type



Here we can see that heavy payload were delivered to low orbits like – ISS, PO, VLEO and they happened to have high success rate. The longer the flight than lower is payload.

Launch success yearly trend



- we see that since time the quality of booster versions has improved and accordingly success rate increasing as well.

EDA with SQL

https://github.com/mkhayom/master/blob/master/W2_EDA%20with%20SQL.ipynb

All launch site names

Task 1

Display the names of the unique launch sites in the space

```
[88]: qr = 'select unique(LAUNCH_SITE) from spacex'
      pd.read_sql(qr, pconn)
```

[88]:

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS SLC-40
2	CCAFSSLC-40
3	KSC LC-39A
4	VAFB SLC-4E

- Find the names of the unique launch sites
- Present your query result with a short explanation here
- We see that we have 5 unique launch site

Launch site names begin with `CCA`

```
qr = """
select LAUNCH_SITE
from spacex
where LAUNCH_SITE like 'CCA%' limit 5
"""
pd.read_sql(qr, pconn)
```

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS LC-40
2	CCAFS LC-40
3	CCAFS LC-40
4	CCAFS LC-40

- Find all launch sites begin with `CCA`
- Present your query result with a short explanation here
- We see that only five rows returned hence this launch site has only 5 flight launched.

Total payload mass

```
qr = """
SELECT  CUSTOMER, sum(PAYLOAD_MASS__KG_) as TOTAL_SUM
FROM spacex
WHERE CUSTOMER = 'NASA (CRS)'
GROUP BY CUSTOMER
"""
pd.read_sql(qr, pconn)
```

	CUSTOMER	TOTAL_SUM
0	NASA (CRS)	45596

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here
- Total payload is 45596 kg

Average payload mass by F9 v1.1

```
qr = """
SELECT  AVG(PAYLOAD_MASS__KG_) as average_payload_mass
FROM spacex
WHERE BOOSTER_VERSION like 'F9 v1.1%'

"""
pd.read_sql(qr, pconn)
```

	AVERAGE_PAYLOAD_MASS
0	2534

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here
- Average payload is 2535 kg

First successful ground landing date

```
52]: qr = """
     SELECT  min(DATE) as First_landing
     FROM spacex
     WHERE LANDING__OUTCOME like 'Success%'

     """
     pd.read_sql(qr, pconn)
```

52]:

	FIRST_LANDING
0	2015-12-22

- Find the date when the first successful landing outcome in ground pad
- Present your query result with a short explanation here
- First successful landing was on December 22, 2015

Successful drone ship landing with payload between 4000 and 6000

```
qr = """
SELECT  BOOSTER_VERSION, PAYLOAD_MASS__KG_ ,LANDING__OUTCOME
FROM spacex
WHERE (PAYLOAD_MASS__KG_ >4000 AND PAYLOAD_MASS__KG_ <6000)
AND LANDING__OUTCOME = 'Success (drone ship)'

"""
pd.read_sql(qr, pconn)
```

	BOOSTER_VERSION	PAYLOAD_MASS__KG_	LANDING__OUTCOME
0	F9 FT B1022	4696	Success (drone ship)
1	F9 FT B1026	4600	Success (drone ship)
2	F9 FT B1021.2	5300	Success (drone ship)
3	F9 FT B1031.2	5200	Success (drone ship)

- List the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here
- Four flight match our query

Total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
]: qr = """
SELECT s.success, f.failure FROM
   (SELECT COUNT( MISSION_OUTCOME ) success FROM spacex WHERE MISSION_OUTCOME like 'Success%' ) s,
   (SELECT COUNT( MISSION_OUTCOME ) failure FROM spacex WHERE MISSION_OUTCOME not like 'Success%' ) f

"""
pd.read_sql(qr, pconn)
```

]:

	SUCCESS	FAILURE
0	100	1

Boosters carried maximum **payload**

```
qr = """
SELECT  BOOSTER_VERSION
FROM spacex
WHERE PAYLOAD_MASS__KG_ =
    ( select max(PAYLOAD_MASS__KG_) FROM spacex )

"""
pd.read_sql(qr, pconn)
```

	BOOSTER_VERSION
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 launch records

```
qr = """
SELECT  MONTHNAME(DATE) as month_name, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM spacex
WHERE year(DATE) = 2015 and LANDING__OUTCOME = 'Failure (drone ship)'

"""
pd.read_sql(qr, pconn)
```

	MONTH_NAME	LANDING__OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
0	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Present your query result with a short explanation here
- In 2015 only two flight were condected.

Rank success count between 2010-06-04 and 2017-03-20

```
qr = """
SELECT  LANDING__OUTCOME, COUNT(*) as Count
FROM spacex
WHERE DATE >= '2010-06-04' and DATE <= '2017-03-20' and LANDING__OUTCOME like 'Success%'
GROUP BY LANDING__OUTCOME
order by Count DESC
"""
pd.read_sql(qr, pconn)
```

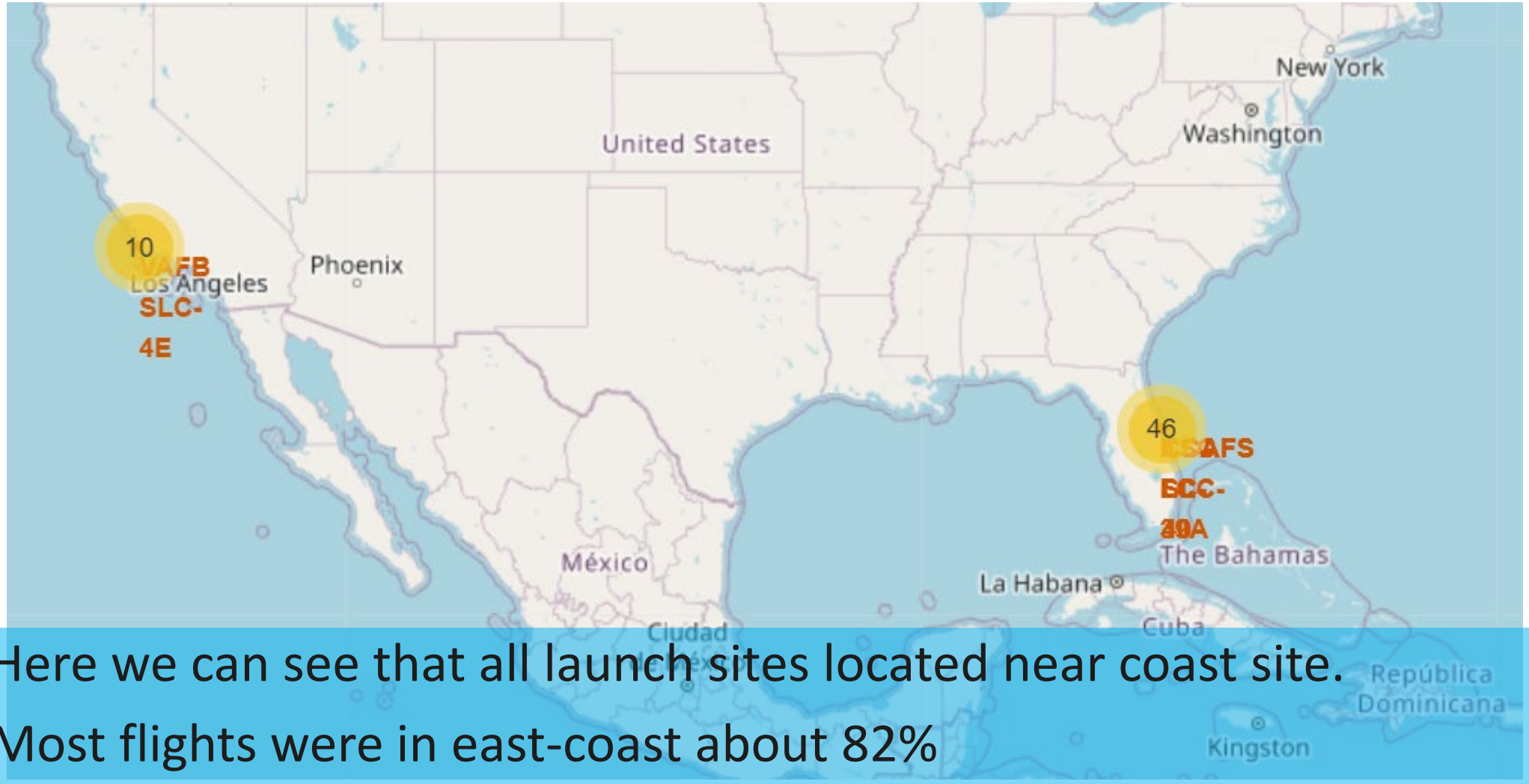
	LANDING_OUTCOME	COUNT
0	Success (drone ship)	5
1	Success (ground pad)	3

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- During this period 5 boosters successfully landed drone ship and 3 boosters on ground pad.

Interactive map with Folium

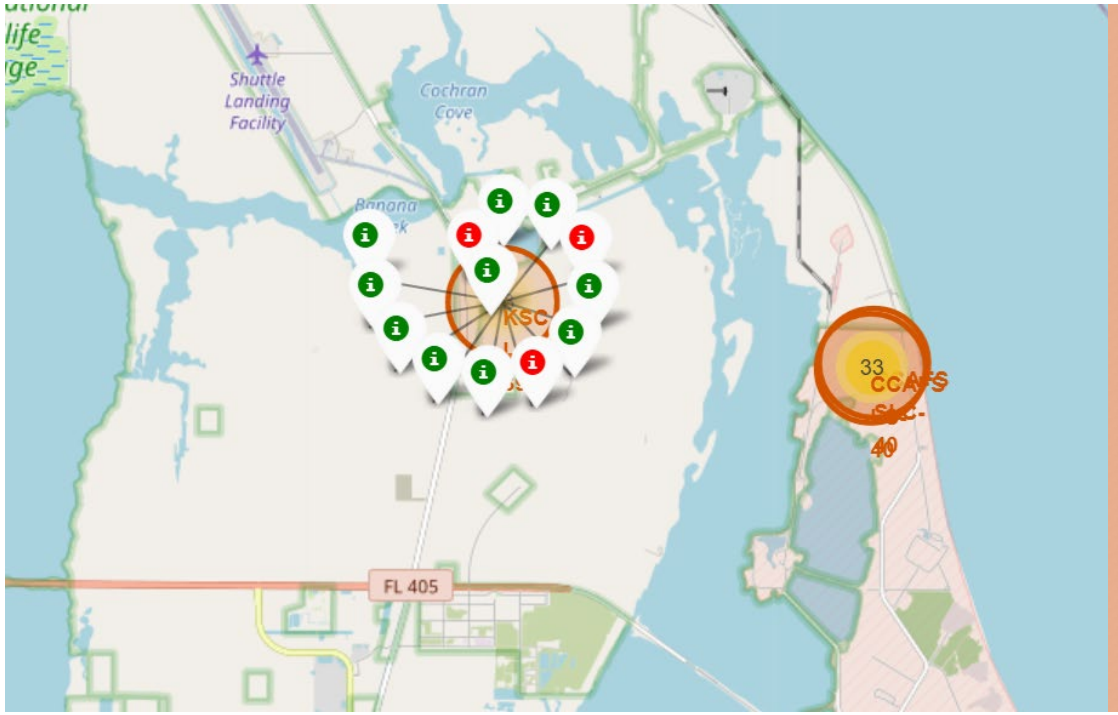
https://github.com/mkhayom/master/blob/master/W3_lab_jupyter_launch_site_location.ipynb

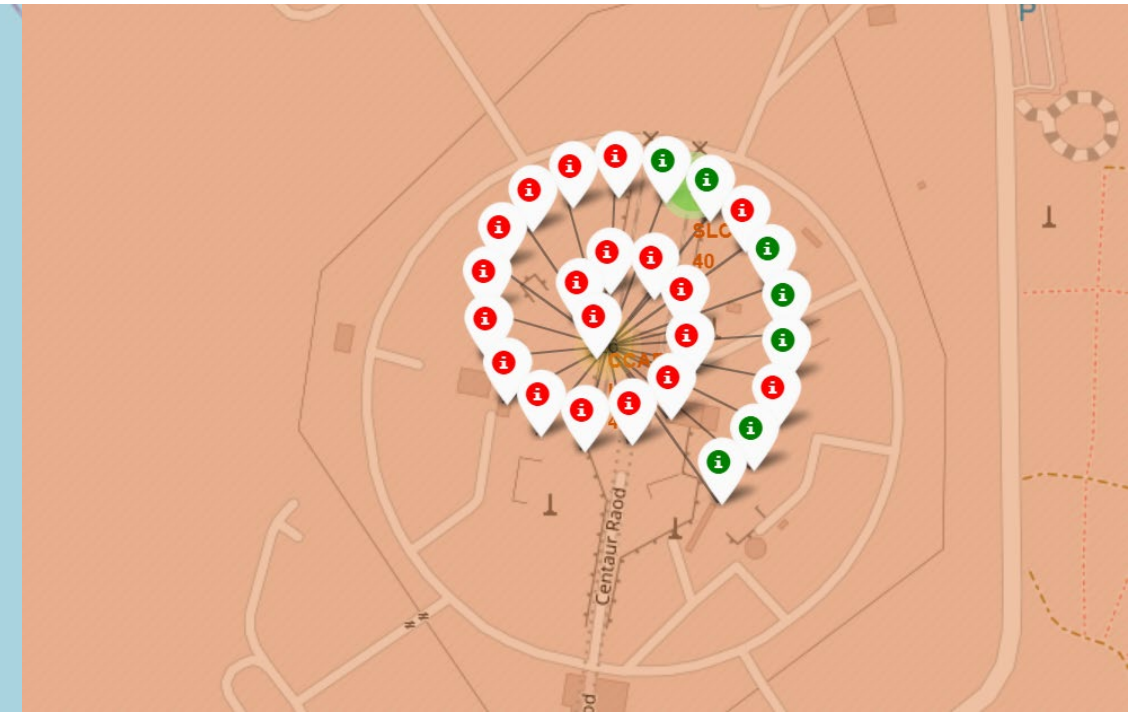
Location of Launch sites



- Here we can see that all launch sites located near coast site.
- Most flights were in east-coast about 82%

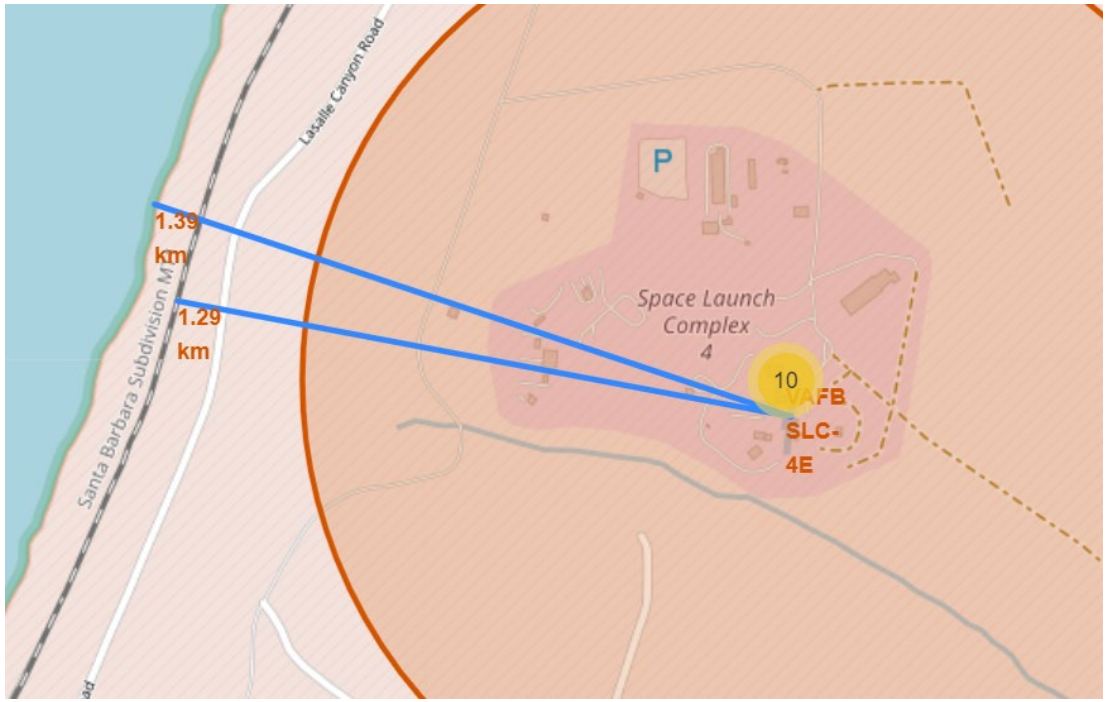
Success/Failed launches





- KSC LC-39A launch site has high success rate. On other hand CCAFS LC-40 has largest number flight and but with Low success rate in the early stages.

Nearest infrastructure

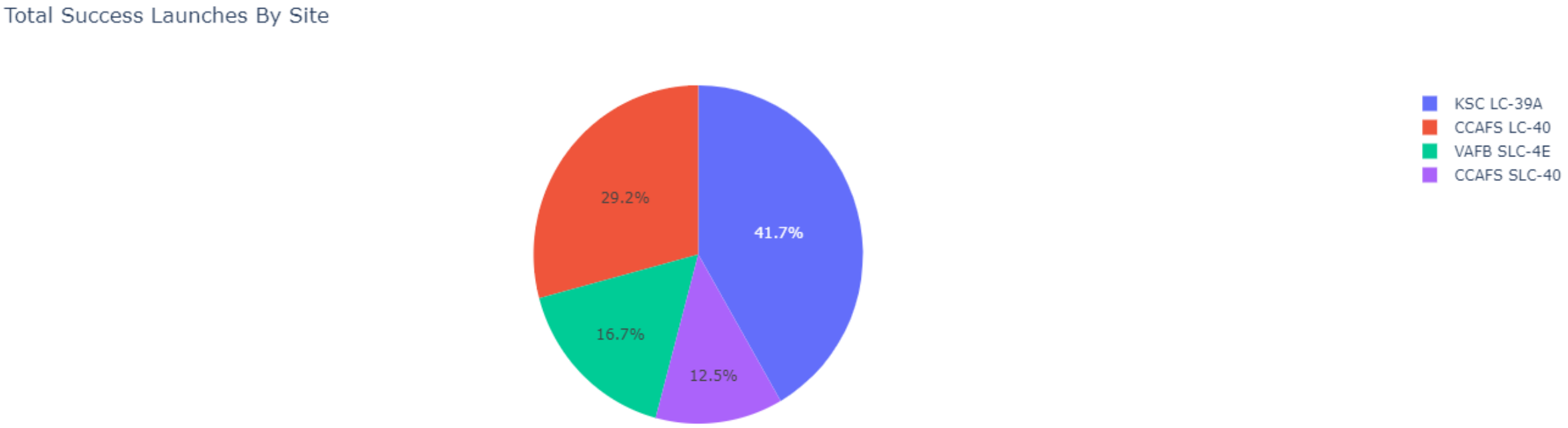


- Here we analyze distance to coastline and railroad in west and east launch sites. The distance to railroad on west is 1.29 km and on east launch site 1.27 with is quite similar, but the distance to coastline differs, 1.39 km on west and 0.85 on east.

Build a Dashboard with Plotly Dash

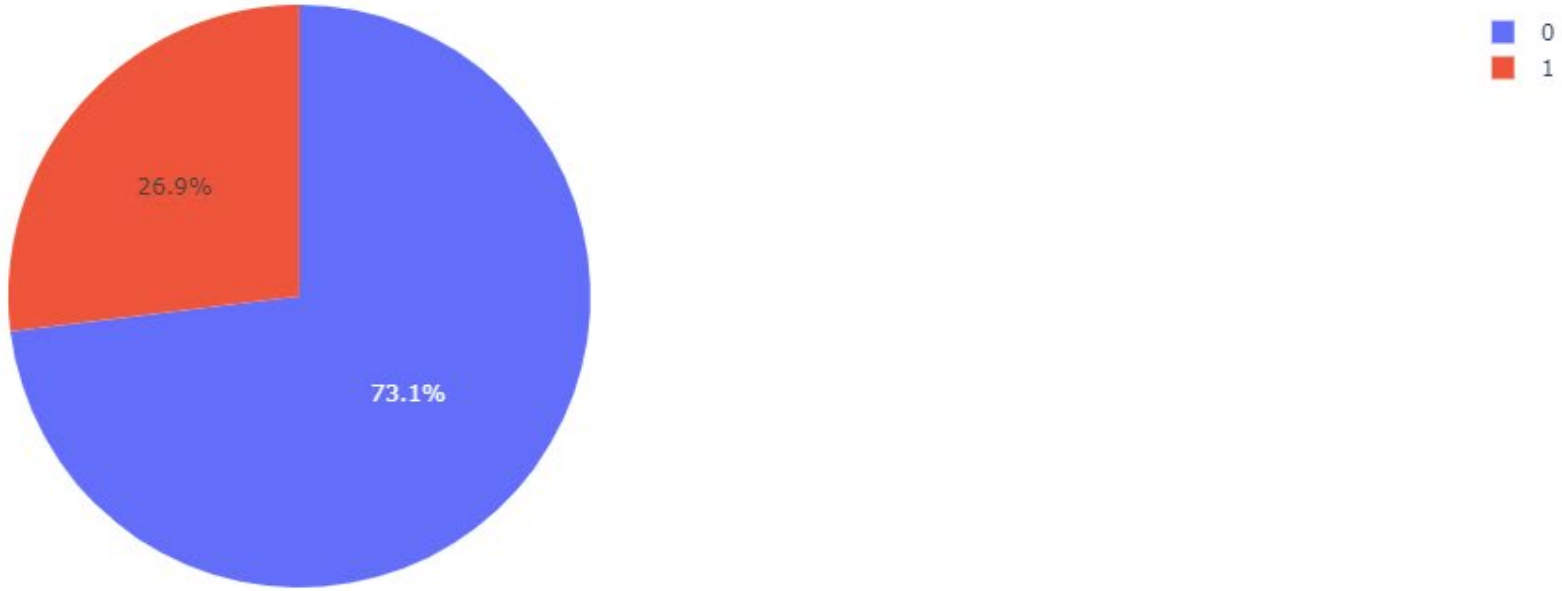
https://github.com/mkhayom/master/blob/master/spacex_dash_app.py

Total Success Launches by Site



- We see that success rate of KSC LC-39A land site significantly higher than other. On second place we can put CCAFS LC-40.

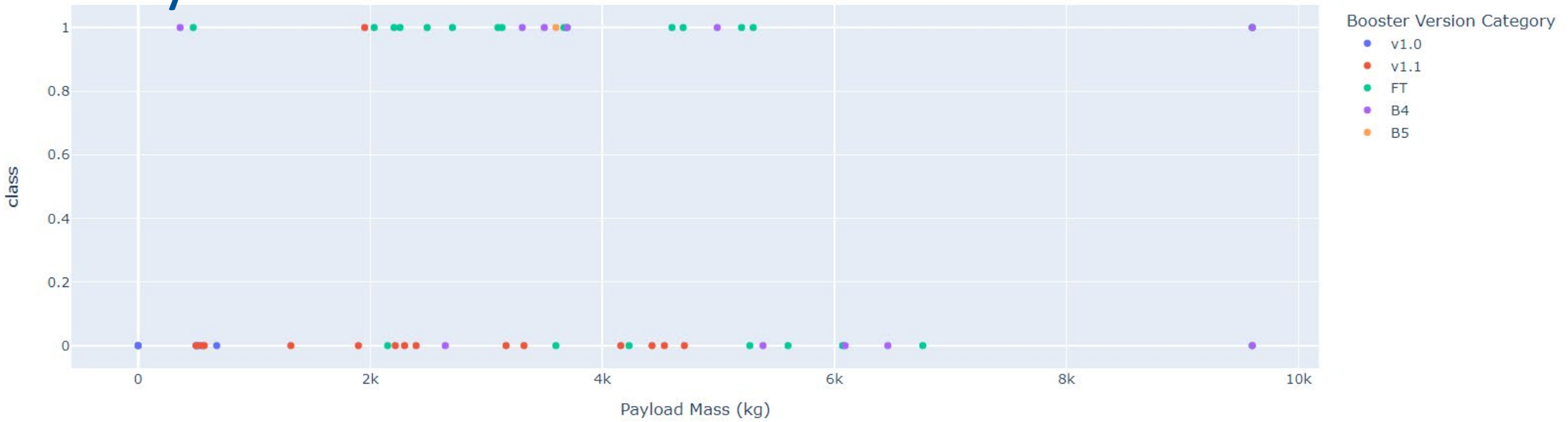
CCAFS LC-40 launch site breakdown



- we can observe that only 27% of flights were successful.

Payload range (Kg):

Payload vs. Launch Outcome



- Here we can observe that booster payload in range [2000-6000] kg have high success rate. From all booster versions “FT” was most successful.

Predictive analysis (Classification)

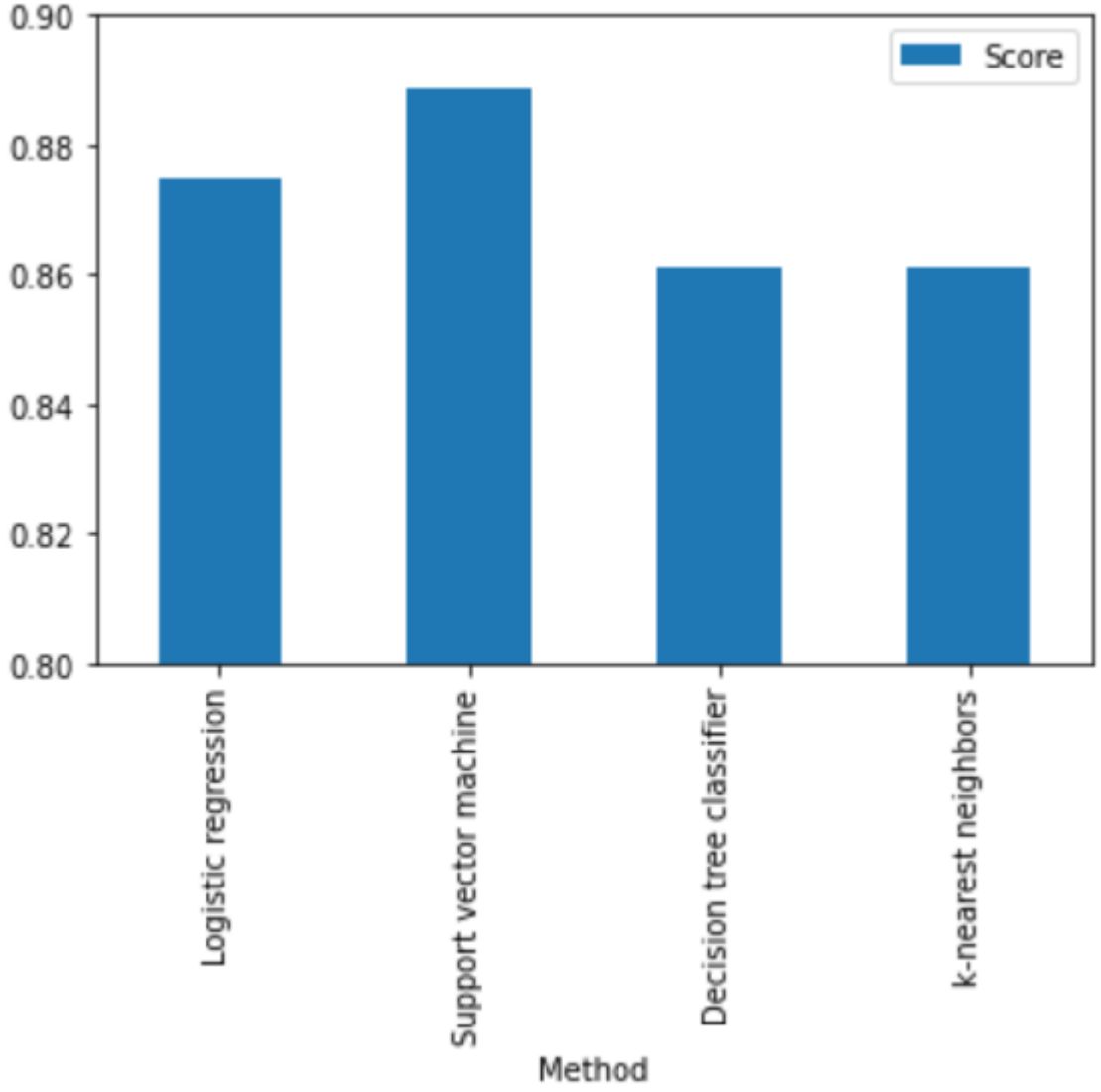
https://github.com/mkhayom/master/blob/master/W4_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Classification Accuracy

Visualize all the built model accuracy for all built models, in a barchart

Find which model has the highest classification accuracy

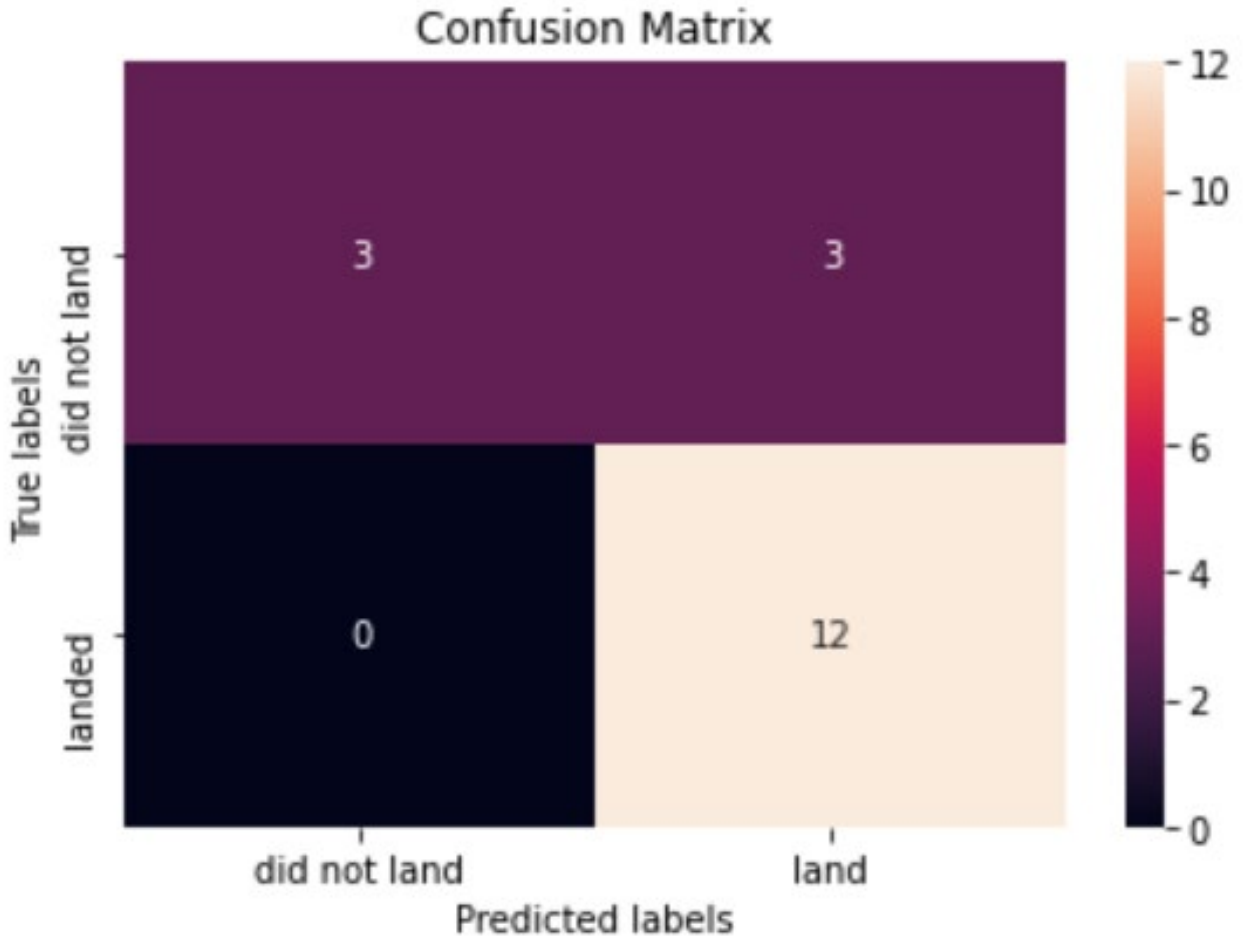
Support Vector Machine object have high predictive score.



Confusion Matrix

Show the confusion matrix of the best performing model with explanation

Support Vector Machine object



CONCLUSION

- KSC LC-39A launch site has most high success rate
- higher earth orbit have high success rate – ESL1, GEO, GTO, HEO, but with low payload carried
- Success rate have positive trend over the years.
- From all booster versions “FT” was most successful.
- Support Vector Machine method have high predictive score; hence we can predict landing outcome with used data with probability of 88%

APPENDIX

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project
- features_one_hot.to_csv('dataset_part_3.csv', index=False)